MARIA REMEMBERS THAT THE LATEST COPY OF FORMAT IS OUT...

...AND CAN'T HELP BURSTING INTO SONG!

## More Than Just A Spoonful Of Sugar

# CONTENTS

Vol.9 N°3.                    November 1995.

### THIS MONTHS ADVERTISERS:-

| | |
|---|---|
| CHEZRON SOFTWARE (OUTLET) | 18 |
| FORMAT (SAM REPAIR SERVICE) | 14 |
| FRED PUBLISHING | Back Cover & 17 |
| KOBRAHSOFT | 18 |
| REVELATION SOFTWARE | 34 |
| SATURN SOFTWARE | 18 |
| S.D.SOFTWARE | 6 |
| WEST COAST COMPUTERS | 2 |

# NEWS ON 4

## EDINBURGH SAM SHOW

SAM fans will be flocking to Ingliston on the outskirts of Edinburgh on Sunday the 12th November for the biggest SAM event north of the border in years.

Scottish Computer Fairs, an Aberdeen based company that has a long record of organizing general computer shows, and this time they have put SAM on the map by making it a feature part of their show, with details being included in all their advertising. The show is on from 10am to 4pm at the Royal Highland Centre, Ingliston, Edinburgh which is close to Edinburgh airport and has excellent motorway links.

Being Scotland the shops are open on Sundays, and the buses running, so why not make it a family day out and send those who don't like computers off into the centre of Edinburgh.

## HARD DRIVE LAUNCHED

Its here at last, the interface that SAM users have been eagerly waiting for, the SAM Hard Disc Interface from S.D.Software. Premiered at the Gloucester show in October, the interface, which allows standard IDE hard drives to be connected to SAM is now shipping. For the very reasonable sum of £60 you get the interface (in a standard white plastic box) and the driving software. All you need to provide is the IDE hard drive with a suitable power supply. Drives as large as 1.2Gb have been tested but it is expected that most people will be able to obtain a second-hand IDE drive from someone forced to upgrade their PC's drive due to the ever increasing demand PC software makes for disc space. Cables to link the drive to the interface are readily available or can be ordered from S.D. direct. There was not time for S.D. to get a new advert for this issue but look out next month.

Included in the price is free software upgrades for the first 12 months so there is nothing to lose by being one of the first to own one of the systems. Without doubt this is one of the most important developments in the SAM world in recent years and software companies seem very keen to support the system.

## SPRITELY SAM

FRED Publishing have launched a new Sprite designer called, wait for it, SAMSprite. OK so it is one of those names that took them simply months to think up, but don't let that put you off.

SAMSprite does all you would expect a Sprite designer to do, it does it well, with good presentation, and at £9.99 (£9 to INDUG members) it does it at a price that everyone can afford. The program allows you to export your creations so that they can be used in Basic, machine code, GamesMaster, C.... It also allows you to view animation within the package so you don't have to waste time flipping between programs while you develop your stunning graphics.

Fred have also announced that the new updated version of the SAM C compiler will be sent out to registered users by the end of November together with news of a new collection of C libraries coming soon.

Despite very heavy fog all over Britain, the 5th Gloucester show on Saturday the 14th October was another storming success. Attendance was about 15% down, partly the fog and partly the dreaded flu bug that seems to have hit the country recently (I was laid low the weekend before the show and only fully recovered with hours to go). Two exhibitors also failed to make it to the show, one, Gus Little, was down with the flu bug and the other, Allan Clarkson of Crashed! who let down very badly by the transport he had arranged to get him (and a couple of others) from the far from sunny Leeds railway station to the very sunny Gloucester - Yep, the sun did shine in Gloucester that day. I can think of far better places to be standing around for four hours, waiting for a lift that never turned up. May I suggest a small contract on the life of the person who let you down Allan - I'm sure missing the Gloucester show warrants that at the very least.

Still, there was much of interest at the show. The 'Bring & Buy' saw an incredible amount of hardware and software change hands. From first thing in the morning to last thing there was a never ending stream of people walking in and out of the hall clutching quantities of goodies - especially Spectrum items which seemed to be in great demand.

S.D.Software were showing the SAM IDE Hard Drive system and taking advanced orders. Version one of the software was nearly finished and the only thing that prevented Nev Young selling interfaces on the day was the late delivery of the final PCBs. As you can expect, there was great interest in the IDE interface and lots of SAM users were very keen to get their orders in for one of the first batch. We will bring you a review of the Hard Disc system next month if all goes well.

While still on the subject of shows, you will find all the details of the Scottish show on the News On 4 page. I hope to be there on the 12th, Fred will definitely be going and so will SD Software. There is a good chance that other SAM companies will attend but it is a little too early to say for definite as I write this.

Well, summer is over, the nights have drawn in. So now is the time to really get down to some real computing. The Spectrum and SAM have long been renowned for their user friendliness when it comes to programming so there is no excuse for anyone to miss out on our Winter Programming Challenge. What we want you all to do is write a small Basic program, and I am being deliberately vague about what I class as a small program because I would not want to exclude something just because it has three lines too many. The program can be of any type, serious or not so serious. The best will be used in a new book of Basic programs, with the authors being paid accordingly, and any that don't make the book will be passed on to John Wase for use in Short Spot. Just put your program on a disc, label it clearly with the title and your name. Write a few notes on what the program does and point out any special features. Then send it to us at the normal address. Closing date is 1st March so you have plenty of time.

Until next month.

*Bob Brenchley*, Editor.

---

# SHORT•SPOT
## YOUR HINTS, TIPS AND PROGRAMMING IDEAS

### Edited By:- John Wase.

Hi folks. It's Short Spot time, and, as usual, there's chaos in the Wase Household. The reason is that for the last two months, Wase has sent Bob files he can't read. And Bob 'aint very pleased, Bob 'aint! More specifically, it's the first file on a SAMdisc that's giving trouble, and there's a good reason for it, too. I eventually got round to asking what the problem was, and now I know, it's so interesting that I thought I would pass on the information and the solution.

Now, to understand just what's going on, I have to tell you a little bit about my systems. A few years ago, I'd save PLUS D Spectrum stuff on a 5¼" disc, and SAM stuff on a 3½" disc. The text used to come on a separate 3½" PLUS D disc. Then, as things changed, a 3½" IBM PC compatible disc, which enabled me to check the program on the PLUS D while I wrote on my laptop. My current setup is a Pentium notebook saving text at first to hard disc, and at the end to three 3½" floppies; two to archive and one to Bob, together with a SAM disc formatted under MasterDOS, which can be written to either by the PLUS D or by SAM. Again, one for me and one for Bob. So contributors please note: there's little point in including short text files on SAMdiscs. Bob currently gets two discs each month - one an IBM PC compatible, with ASCII code on and the other with a mixture of SAM and Spectrum programs. And this latter is the cause of all the trouble.

You see, it's much quicker and easier to copy things using MasterDos. It's quick and flexible. But, there's this problem...

Now, when Dr. Wright wrote MasterDOS, he hadn't quite grasped all the implications. You can, for instance, guarantee that if you devise a system, someone else will foul it up. And you're right! Not appreciating what clowns like Wase might get up to, Dr. Wright assumed that the first file would always be a SAM file. Indeed, since we all know that the first file on a SAM disc is likely to be the boot file, you might well expect it to be MasterDOS. Well, of course, in my case, it wasn't.

When you look carefully at it, it works fine for a SAM file. Dammitt, it ought to oughn't it, for it's a formatted disc for SAM! But if the first and only the first file is a Spectrum file, then this is locked solid. You can't recover it subsequently. As Bob has found. And this is because there is a little area there in the directory track which is clear in MasterDOS files, but not in Spectrum files. And it is to this little area IN THE FIRST FILE POSITION ONLY that the System writes the name of the disc. All over the header of your Spectrum file. Yeah, man; tramps all over it! No wonder you can't get it off again.

The quick fix is to type 10 GOTO 10 <Enter>. Now SAVE "Dummy". So the first directory entry is a dummy. And after that, everything works OK, because the name of the disc is written in an unused bit of the header for the SAMfile "Dummy". Then you can save Spectrum files to your heart's content. AND

recover them intact. Don't forget: you read it in this column first!!!

Here's one of Roy Burford's files that suffered in this way - it should have been in the October issue. The lottery program of G.J.Symes (September's Short Spot) has given Roy Burford a problem. He converted it to a Commodore 64 (Boo! Hiss!), but when he tested it, found that he got occasional groups of numbers which were not in ascending order, and, indeed, occasionally got two identical numbers. So he tested the Spectrum version even more thoroughly, and found a similar result: one group wrong approximately every three screenfuls. Roy therefore analysed the program and came up with an answer which produces groups of numbers in ascending order without fail, as far as Roy can determine.

The main trick is the use of the variable 'c' in line 120, and rejecting values of a$ which are less than or equal to the a$ last concatenated to n$. The program is shorter than G.J.Symes', too. Without REMs it is 400 bytes against Symes 422 bytes, though he, of course, had the original idea.

Here it is, then: "LOTTERYRB". Get typing!

```
  1 REM Vol.8 No12. FORMAT. Aug
    ust 1995. Short Spot. p10.
  2 REM Revised on ZX Spectrum+
    128K by B.C.R.Burford 22089
    5.
  3 REM Original form not consi
    stently producing rising se
    t of numbers. Re-revised 24
    0895.
 10 REM Lottery by G.J.Symes
 20 DIM a$(4): LET c=.1
 25 GOSUB 120
 30 LET n$=a$: FOR c=1 TO 4: GO
    SUB 120
 40 IF a$<=n$(LEN n$-3 TO ) THE
    N GOSUB 120: GOTO 40
 50 LET n$=n$+a$: NEXT c
 70 PRINT "Your lottery numbers
```

```
    are:""     ";n$
 90 GOTO 20
120 LET a$=STR$ INT (RND*(c*10+
    9)+1): IF a$(2)=" " THEN LE
    T a$="0"+a$
130 RETURN
```

As you might have noticed, this piece was missing from October's Short Spot, having fallen victim to the SAM/Spectrum-file-first bug, but Bob inadvertently left the next and related piece in: sorry to those readers who were confused. Here's that second bit again in its proper place.

Just to turn the tables, Ettrick Thomson's come up with a whole raft of ideas on improving Roy's recent program; 'rounding', so I've sent them on. No doubt we'll get some nicely rounded ideas coming back from Roy in due course. Many thanks to you both.

Let's continue with Miles Kinloch; again a bit which the Bug squeezed out of September's Short Spot. Earlier on, he sent me a letter which continued his DOSPEEK theme for the PLUS D. He's just made an execute file to print out the DOS settings 0-11, with a brief descriptive label for each. It also displays which DOS is installed; G+DOS or BetaDOS. Unfortunately, it won't work with UniDOS and it won't work on DISCiPLE, only the PLUS D, since it's assembled to execute at address 15318, the PLUS D's disc buffer area. Being in the form of an execute file gives it the advantage of not disturbing the Spectrum RAM, so it can conveniently be used while another program is in memory. The program on the disc POKEs and then saves the execute file which is itself saved at P2.

Here it is—

```
 10 REM      DOS SETTINGS
 20 REM  (PD) 1995 M. Kinloch
 30 REM      (FOR PLUS D ONLY)
 40 REM
 50 CLEAR 49999: PRINT "PLEASE
```

```
    WAIT...": LET c=0: FOR a=50
    000 TO 50457
 60 READ d: LET c=c+d: POKE a,d
    : NEXT a: CLS
 70 IF c<>36282 THEN PRINT "ERR
    OR IN DATA!": STOP
 80 PRINT "PRESS ANY KEY TO SAV
    E": PAUSE 0: CLS
 90 SAVE d1"DOSSET.EXE"X,50000:
    STOP
100 DATA 215,107,13,62,2,215,1,
    22
110 DATA 17,232,59,6,26,205,217
    ,60
120 DATA 24,25,190,64,32,124,32
    ,86
130 DATA 65,76,85,69,32,124,32,
    32
140 DATA 32,32,70,85,78,67,84,7
    3
150 DATA 79,78,13,6,32,62,45,21
    5
160 DATA 16,0,16,249,62,13,215,
    16
170 DATA 0,6,12,33,0,32,197,229
180 DATA 62,12,144,245,254,10,4
    8,5
190 DATA 62,32,215,16,0,62,32,2
    15
200 DATA 16,0,62,32,215,16,0,24
    1
210 DATA 6,0,79,215,43,45,215,2
    27
220 DATA 45,17,65,60,6,7,205,21
    7
230 DATA 60,24,6,32,32,32,124,3
    2
240 DATA 32,225,126,35,229,245,
    254,10
250 DATA 48,5,62,32,215,16,0,24
    1
260 DATA 245,254,100,48,5,62,32
    ,215
270 DATA 16,0,241,6,0,79,215,43
280 DATA 45,215,227,45,62,32,21
    5,16
290 DATA 0,62,32,215,16,0,62,12
    4
300 DATA 215,16,0,62,32,215,16,
    0
310 DATA 225,193,205,226,60,16,
    143,62
320 DATA 13,215,16,0,62,13,215,
    16
330 DATA 0,17,153,60,6,26,205,2
```

```
    17
340 DATA 60,24,25,68,79,83,32,6
    7
350 DATA 85,82,82,69,78,84,76,8
    9
360 DATA 32,73,78,83,84,65,76,7
    6
370 DATA 69,68,58,32,58,172,48,
    254
380 DATA 68,40,4,254,205,32,7,1
    7
390 DATA 203,60,6,8,24,21,17,21
    1
400 DATA 60,6,6,24,14,66,69,84
410 DATA 65,68,79,83,13,71,43,6
    8
420 DATA 79,83,13,5,200,26,19,2
    15
430 DATA 16,0,24,247,197,229,62
    ,12
440 DATA 144,33,0,61,190,35,40,
    2
450 DATA 24,250,126,254,128,40,
    8,229
460 DATA 215,16,0,225,35,24,243
    ,225
470 DATA 193,201,0,231,219,13,1
    28,1
480 DATA 68,82,73,86,69,32,49,3
    2
490 DATA 84,82,65,67,75,83,13,1
    28
500 DATA 2,68,82,73,86,69,32,50
510 DATA 32,84,82,65,67,75,83,1
    3
520 DATA 128,3,83,84,69,80,80,7
    3
530 DATA 78,71,32,82,65,84,69,1
    3
540 DATA 128,4,195,85,83,69,68,
    13
550 DATA 128,5,67,72,65,82,83,3
    2
560 DATA 80,69,82,202,13,128,6,
    84
570 DATA 79,75,69,78,32,69,88,8
    0
580 DATA 65,78,83,73,79,78,13,1
    28
590 DATA 7,202,83,80,65,67,73,7
    8
600 DATA 71,13,128,8,202,70,69,
    69
610 DATA 68,83,13,128,9,76,69,7
    0
```

```
620 DATA 84,32,77,65,82,71,73,7
    8
630 DATA 13,128,10,35,96,127,32
    ,65
640 DATA 83,32,71,82,65,80,72,7
    3
650 DATA 67,83,13,128,11,80,79,
    82
660 DATA 84,32,69,78,65,66,76,6
    9,13,128
9999 SAVE d1"DOSSETTING" LINE 10
```

Many thanks, Miles: you've come up with the goods again!

Now, for those patient people who've been waiting for something for SAM, I have a long snippet from David Foster of Corby, Northants, who writes about the comments I made on SAM's 'GET A' command.

Firstly, he writes, the command is acting just as it was meant to do. In David's copy of the SAM user guide (page 84), it states that GET A returns values of 0-9 for the numeric keys and 10 for A, 11 for B, etc. This command was apparently intended for use when selecting an option on a menu. Unfortunately, neither the user guide nor the SAM Technical Manual contains a table of the values returned for all the keys. In point of fact, each key can return up to four different values, depending on whether it's used on its own, or at the same time as SHIFT, SYMBOL or CONTROL. Alphabetic keys return the same value whether used on their own, or at with SHIFT. There is some rather arbitrary and odd duplication in the values returned. For instance, the value 9 is returned by 9, SHIFT-2 or SYMBOL-L. A further complication is that the values returned by GET A seem at first unrelated to those returned by CODE INKEY$, but after some experimenting, David found the following relationship:

With two sets of exceptions, adding either 169 or 201 modulus 256 to the value returned by CODE INKEY$ gives the GET A code. So, if CODE INKEY$ is in the range....

| | |
|---|---|
| 0 to 31, | add 201 |
| 32 to 63, | add 169 |
| 64 to 95, | add 201 |
| 96 to 127, | add 169 |
| 128 to 159, | add 201 |
| 160 to 191, | add 169 |
| 192 to 223, | add 201 |
| 224 to 255, | add 169 |

Exceptions...

If CODE INKEY$=6, 16 or 22, GET A will wait for another key. This seems to be due to CAPS=6.

If CODE INKEY$ is in the range 48 to 57, add 208 mod 255: this gives values of 0 through to 9 for the numeric keys.

Evidently the reason for adding 169 or 201 is so that the alphabetic keys give the same result when pressed on their own, or with the SHIFT key.

Finally, writes David, it's worth mentioning that CODE INKEY$ and GET A use a table of values stored at address 22752 (hex 58E0). This table can be modified using the KEY command, e.g. KEY 84, 250 will amend the table so that SHIFT-TAB returns a different value from TAB on its own; useful if you want to use SHIFT-TAB as a back tab key in a text editor.

Many thanks for clearing this up, David.

Ben Booth of Cheltenham also writes in less detail on the same subject, and again refers interested readers to the SAM user's guide, so it's clear that some of our readers do know quite a lot about this. Ben also mentions, however, that there is a problem over discs sent through the post. The Royal Mail has started using an electrostatic post code sorter. This machine has the antisocial habit of completely wiping discs of all data. Royal Mail's advice is to make sure you keep backups of any discs you post. Readers, beware!

Ben also has sent me a couple of programs to type in. The first is PI-GUESS, and is interesting, for it is based on a practice exam question which Ben was asked to do. This asked one to work out a method of obtaining pi by using a computer procedure which generated two random numbers between -1 and +1. After doing the question, Ben decided to write the program to check if the method worked. The program uses the two random numbers to represent a point. The point must be in a square of area 4 units (between-1 and +1 on each axis). The program then uses the equation of a circle to see if the point is on a circle, centre the origin, and radius 1. The area of the circle should be pi. By looking at how many points land on the circle, compared with the square, the computer can have a guess at pi. Although this is intrinsically not a very exciting program, it is interesting to watch SAM gradually getting nearer the real value... The results also suggest that when push comes to shove, SAM's random numbers aren't terribly random. Type it in and have a look.

```
 10 REM Program to calculate
    pi using random numbers
 20 REM Ben Booth, 1995
 30 LET a=0, b=0
 40 setgraph
 50 :
 60 DO
 70 LET x=(RND*2)-1
 80 LET y=(RND*2)-1
 90 LET a=a+1
100 IF (x^2)+(y^2)<=1 THEN LET
    b=b+1
110 LET p=(4*b)/a
120 PRINT AT 0,0;"x= ";x
130 PRINT AT 1,0;"y= ";y
140 PRINT AT 2,0;"a= ";a
150 PRINT AT 3,0;"b= ";b
160 PRINT AT 5,0;"pi= ";p
165 PLOT x,y
170 LOOP
180 :
190 DEF PROC setgraph
```

```
200 PEN 3
210 LET xrg=5.1,yrg=3.8
220 LET xos=2.5, yos=1
230 PLOT 0,0: PLOT -1,-1: DRAW
    TO -1,1: DRAW TO 1,1: DRA
    W TO 1,-1: DRAW TO -1,-1
240 CIRCLE 0,0,50
250 PEN 7
260 END PROC
```

After all that stress, it's nice to have a little relax. Again, on the theme of random numbers, this snippet simply uses random numbers to generate landscape pictures. Save them, and have a little screenshow afterwards: it can be quite relaxing to sit in front of the machine and just flick through all these landscapes...

Many thanks for those programs, Ben.

```
 10 REM landscape generator
 20 PEN 4
 30 LET b=10, xrg=250
 40 LET r1=INT (RND*4)
 50 LET r2=INT (RND*16)
 60 PLOT 0, INT (RND*b)+1
 70 DRAW 30, INT (RND*b)+1
 80 DRAW 30, INT (RND*b)+1
 90 DRAW 30, INT (RND*b)+1
100 DRAW 30, INT (RND*b)+1
110 DRAW 30, INT (RND*b)+1
120 DRAW 30, INT (RND*b)+1
130 DRAW 30, INT (RND*b)+1
140 DRAW 30, INT (RND*b)+1
150 DRAW 7, INT (RND*b)+1
160 FILL 0,0
170 IF r1=1 THEN PEN 0: GOTO 1
    90
180 PALETTE 1,r2+16: PEN 1
190 : FILL 200,150
200 IF r1=1 THEN PALETTE 6,15:
    GOTO 220
210 PALETTE 6, INT (RND*16)+96
220 PEN 6 : LET c1= INT (RND*2
    00), c2=INT (RND*100)+50,
    c3=INT (RND*15)+10
230 CIRCLE c1,c2,c3
240 FILL c1,c2
250 PAUSE : RUN
```

Let's stay with SAM for a while. John McCabe of Fetcham, Surrey, mentions that when you wire a program that sends

control codes to the printer, it is usual to type, for example...

PRINT #4; CHR$ 27;"x1"; CHR$ 27; "E"; CHR$ 27;"M"

It would be much easier to type...

PRINT #4;"Esc x1 Esc E Esc M" (Spaces are for clarity)

Well, CHR 27 could be assigned to a key and that would make things a bit shorter, but it would be printed as a question mark. The solution is to use CHR$ 155 which also acts as the escape code (not many people realise this). It can be designed like any other graphics character thus...

POKE 21864,192,152,208,155,202,2 6,3,0

John likes to use it on Symbol E, so do KEY 191,155. John suggests a similar approach might work on the Spectrum if one uses UDG character 155. C'mon folks, give it a whirl and let me know if it does!

Many thanks, John.

Back to the Spectrum, now, and Miles Kinloch of Edinburgh once again has sent me an invaluable disc. Miles has just had an idea, and being Miles, it's a good 'un. Hands up if you've got Tasword 2. Yes, thought you had. And if you've struggled, trying to convert Tasword 128 opentype files to Tasword 2 format. Stuff it, I hear you cry. Yes; I know you've got the well-known utility T3TO2, but have you ever timed it? Being in Basic, it plods along like the proverbial glacial slug! At this point, Miles came out of his reverie, and re-invented the wheel, but assembled as code. Wow! The code wheel sure is a fast wheel. Try translating the Tasword 'Tutor' file - it's 17K, takes T3TO2 the better part of an hour, but TAS128TO2 manages it in a mere 12

seconds! Here it is...

```
10 REM FAST FILE CONVERTER FOR
20 REM TASWORD 128 > TASWORD 2
30 REM (PD) Miles Kinloch 1995
40 REM
50 CLEAR 31899: GOSUB 1000
60 INPUT "Load from which driv
   e? ";s: IF s<>1 AND s<>2 TH
   EN GOTO 60
70 INPUT "Save to which drive?
   ";d: IF d<>1 AND d<>2 THEN
   GOTO 70
80 CAT s;"????????_T":: LET n$
   ="          _T": INPUT "Filen
   ame ? "; LINE n$( TO 8)
90 CLS : CLEAR #: OPEN #5;ds;n
   $IN
100 RANDOMIZE PEEK 23631+256*PE
   EK 23632+PEEK 23584+256*PEE
   K 23585+30: POKE 31915,PEEK
   23670: POKE 31916,PEEK 236
   71: POKE 23729,0
110 LET end=USR 31900: CLOSE #"
120 SAVE dd;n$( TO 8)CODE 32000
   ,end-32000: STOP
1000 REM CODE POKER
1010 LET c=0: FOR n=31900 TO 319
   83: READ d: LET c=c+d: POKE
   n,d: NEXT n
1020 IF c<>8178 THEN PRINT "ERRO
   R IN DATA!": STOP
1030 RETURN
2000 DATA 62,5,205,1,22,30,0,33,
   0,125,34,254
2010 DATA 124,237,75,0,0,120,177
   ,68,77,200,213,229
2020 DATA 205,230,21,225,209,254
   ,13,32,10,87,58,177
2030 DATA 92,183,122,40,15,24,32
   ,254,10,40,219,119
2040 DATA 35,28,175,50,177,92,24
   ,210,123,254,64,32
2050 DATA 8,175,95,60,50,177,92,
   24,197,54,32,35
2060 DATA 28,24,237,6,64,54,32,3
   5,16,251,24,182
9999 SAVE d1"TAS128TO2" LINE 10
```

Miles also includes a further snippet to keep us happy. It's called SNAP OUT, and Miles has sent it before, only it seems to have got lost in all the upheaval. It is a PLUS D DOS temporary overlay providing a snapshot

key to force an exit to Basic (similar to the Multiface option - but better!). The only essential criteria are that a) the Basic area must be intact and b) the system variables must still be viable. Factors such as a non-Basic-friendly interrupt routine (one of the most common causes of failure with the Multiface option, especially with games) or the 128K system variables in the printer buffer having been overwritten, are no obstacle to SNAP OUT.

The reason it should be treated as a temporary overlay, rather than a permanent patch, is that it overwrites two DOS commands: FORMAT and SAVE SCREEN$. This is necessary in order to achieve BetaDos compatibility, since the latter uses every byte of PLUS D RAM, leaving none to spare.

SNAP OUT takes the form of an execute file, which is created and saved by the Basic installation program. Full information is given on-screen; if you don't want to type in all the information, then you'll just have to keep Format carefully, won't you!

```
10 REM          SNAP OUT
20 REM     By Miles Kinloch
30 REM          (PD) 1994
40 REM
60 CLEAR 49999: GOSUB 1000
70 PRINT TAB 11; BRIGHT 1;" SN
   AP7"
80 PRINT ' TAB 6;"FOR G+DOS OR
   BETADOS"
90 PRINT '''"Have you ever wish
   ed the PLUS D had an 'Exit'
   option like the Multi-face
   , for escaping to BASIC fro
   mprograms where the Break k
   ey is inoperative? This uti
   lity may bethe answer - it
   quite literally lets you 's
   nap out of them' by pressi
   ng the snapshot button,   t
   hen key 'E'."
100 PRINT ''"As it overwrites th
   e FORMAT and SAVE SCREEN$ c
   ommands, it shouldonly be r
```

egarded as a temporary over lay rather than a permanent feature. Install it, there fore, only when required."
```
110 PRINT #0; AT 1,9;"PRESS ANY
   KEY": PAUSE 1: PAUSE 0: CL
   S
120 PRINT ''"SNAP7loads in the f
   orm of anExecute file. It i
   ntegrates withthe DOS in a
   special way which  ensures
   that, if there are any  oth
   er DOS modifications alread
   y present which themselves
   alter  the snapshot routine
   s,these willnot be affected
   and will work asnormal. SN
   AP7can therefore  live quit
   e happily alongside thelike
   s of Pick-POKE-It, PLUS D P
   oker,the key 'O' Interrupt
   patch etc."
130 PRINT ''"Even if you already
   have a Multi-face, the SNA
   P7function is actually more
   powerful in some  ways, an
   d can succeed in certainsit
   uations where the Multiface
   'Exit' would fail."
140 PRINT #0; AT 1,9;"PRESS ANY
   KEY": PAUSE 1: PAUSE 0: CL
   S
150 PRINT ''"For instance, where
   a program  has been loade
   d from 128 BASIC  and the 1
   28K system variables inthe
   printer buffer have become
   overwritten. The Multiface
   Exit would then simply cra
   sh, but theSNAP7routine wil
   l detect if this is the cas
   e, and exit to 48BASIC inst
   ead. So, if you find  yours
   elf in 48K mode after exit-
   ing a 128K program, this wi
   ll bethe reason."
160 PRINT ''"Another situation w
   here SNAP OUTcan cope is wh
   en the 128K shadowROM is pa
   ged in. Here, the Multi-fac
   e does not offer the 'Exit'
   option at all!"
170 PRINT #0;AT 1,9;"PRESS ANY
   KEY": PAUSE 1: PAUSE 0: CLS
180 PRINT ''"Finally, if the pro
```

gram is running an Interrup
t Mode 2 routine, this will
be turned off before exiti
ng, in case it isn't compat
-ible with the BASIC editor
. This is particularly true o
f 128 modewhere RAM paging
can be an addedproblem for
Interrupt routines -yet ano
ther instance where SNAP OU
T may succeed where the Mul
ti-face fails."

```
190 PRINT ''"SNAP7can also be ve
    ry handy for escaping from
    programs where23728 has bee
    n POKEd for error-trapping.
    "
200 PRINT ''"TO SAVE THE 'SNAP O
    UT' EXECUTE FILE, PUT A DI
    SC IN DRIVE 1 AND PRESS 'S'
    NOW. ANY OTHER KEYRE-READ
    THIS INFO.": PAUSE 1: PAUSE
    0
210 IF INKEY$="s" OR INKEY$="S"
    THEN CLS : SAVE d1"SNAP OU
    T.X"x,50000: PRINT "LOAD d1
    ""SNAP OUT.X""xINSTALL.": P
    AUSE 1: PAUSE 0
220 RUN 70
1000 REM CODE POKER
1010 LET c=0: FOR n=50000 TO 501
     38: READ d: LET c=c+d: POKE
     n,d: NEXT n
1020 IF c<>14996 THEN PRINT "ERR
     OR IN DATA!": STOP
1030 RETURN
2000 DATA 167,237,91,162,32,33,1
     85,42,237,82
2010 DATA 200,33,185,42,34,162,3
     2,54,33,35
2020 DATA 115,35,114,35,17,9,60,
     235,1,51
2030 DATA 0,237,176,17,38,48,1,3
     7,0,237
2040 DATA 176,175,33,200,32,119,
     35,119,35,119
2050 DATA 201,229,6,251,237,104,
     203,85,192,253
2060 DATA 33,58,92,253,54,0,255,
     253,54,49
2070 DATA 2,253,203,1,102,196,45
     ,48,205,215
2080 DATA 34,205,39,48,237,123,6
     1,92,253,203
2090 DATA 1,102,32,1,225,251,33,
```

```
3,19,195
2100 DATA 79,0,201,175,237,71,23
     7,86,201,58
2110 DATA 92,91,230,32,33,0,91,6
     ,20,134
2120 DATA 35,16,252,254,222,40,4
     ,253,203,1
2130 DATA 166,62,16,50,92,91,195
     ,63,34
9999 SAVE d1"SNAP OUT" LINE 10
```

And that's all for now. The next issue
is Christmas. It's almost too late for
Christmas goodies, but if you have any
REALLY SHORT ones and can get them
to me quickly, then I need them. I've got
nothing Christmassy at the moment...
Please send them to:-

John Wase,
Green Leys Cottage,
Bishampton,
Pershore,
Worcs,
WR10 2LX.

See you next month.

# FASTFILE
## Sorting & Printing

Part 4.                    New Routines for *FASTFILE* By:- Cliff Jackson.

If you have typed in last months
routines then we are now ready to get
down to the real job of printing from
*Fastfile*. Enter the following program
and save as "b$printer" LINE 9000 but
omit the REM lines. When run it will
auto-load both "b$alpha" and "b$sortC",
which we have already created. Take
care to distinguish between 1 and L and
between 0 and o - L and o are variables
declared in line 9020. These have been
used in places to save memory but NOT
in the lines which do the printing as that
would slow the printer.

```
10 CLS : PRINT AT VAL "10",VAL
   "6";"Prepare printer";#o;"
         THEN PRESS A KEY": PAU
   SE o: CLS
12 LET q=PEEK VAL "23627"+VAL
   "256"*PEEK VAL "23628"+VAL
   "6"
20 POKE VAL "65381",INT (q/VAL
   "256"): POKE VAL "65380",q
   -(PEEK VAL "65381")*VAL "25
   6"
30 FOR x=VAL "65382" TO VAL "6
   5393": POKE x,o: NEXT x: LE
   T w=VAL "60": POKE VAL "653
   90",w
40 RANDOMIZE USR VAL "65400"
60 LET v=o: LET page=L
100 POKE #5,VAL "132": POKE #6,
    L
110 LPRINT CHR$ VAL "27";CHR$ V
    AL "15";
120 POKE @6,o
130 LPRINT
140 LET a=o: LET tab=L
1001 PRINT AT VAL "10",VAL "10";
     "Page ";page: LET p=PEEK VA
     L "65392"+VAL "256"*PEEK VA
     L "65393": IF p=o THEN GOTO
```

```
     VAL "1010"
1002 LET w=PEEK VAL "65392": LET
     v=INT (w/INT PI): LET u=w-
     (v*INT PI): IF w=v*INT PI T
     HEN GOTO VAL "1004"
1003 LET v=v+L: LET w=v: LET v=I
     NT PI-u: GOTO VAL "1005"
1004 LET w=v: LET v=o
1005 POKE VAL "65390",w: RANDOMI
     ZE USR VAL "65406"
1010 LET x=PEEK VAL "65382"+VAL
     "256"*PEEK VAL "65383"+L
1020 LET y=PEEK VAL "65384"+VAL
     "256"*PEEK VAL "65385"+L
1030 LET z=PEEK VAL "65386"+VAL
     "256"*PEEK VAL "65387"+L
1050 LPRINT CHR$ PEEK x;: LET x=
     x+1: IF PEEK x<>58 THEN GOT
     O 1050
1060 LET x=x+L: LET tab=tab+37:
     LPRINT TAB tab;
1070 LPRINT CHR$ PEEK x;: LET x=
     x+L: IF PEEK x<>226 THEN GO
     TO 1070
1080 LET x=x+L: LET tab=tab+7: L
     PRINT TAB tab;
1085 IF p=1 AND w=1 THEN LPRINT
     : GOTO 8000
1090 LPRINT CHR$ PEEK y;: LET y=
     y+L: IF PEEK y<>58 THEN GOT
     O 1090
1100 LET y=y+L: LET tab=tab+37:
     LPRINT TAB tab;
1110 LPRINT CHR$ PEEK y;: LET y=
     y+L: IF PEEK y<>226 THEN GO
     TO 1110
1120 LET y=y+L: LET tab=tab+7: L
     PRINT TAB tab;
1125 IF a>=w-v THEN GOTO 1170
1130 LPRINT CHR$ PEEK z;: LET z=
     z+1: IF PEEK z<>58 THEN GOT
     O 1130
1140 LET z=z+L: LET tab=tab+38:
     LPRINT TAB tab;
1150 LPRINT CHR$ PEEK z;: LET z=
     z+L: IF PEEK z<>226 THEN GO
```

```
      TO 1150
1160 LET z=z+L
1170 LPRINT : LET a=a+L: LET tab
     =L: IF a<w THEN GOTO 1050
1510 IF w<>VAL "60" THEN GOTO VA
     L "8000"
1520 LET a=o: LET page=page+L
1530 POKE VAL "65382",PEEK VAL "
     65388": POKE VAL "65383",PE
     EK VAL "65389": RANDOMIZE U
     SR VAL "65406": PRINT AT VA
     L "10",VAL "5";"Insert next
      page (";page;")";#0;"
      THEN PRESS A KEY": PAUSE o
     : CLS : GOTO VAL "1000"
8000 CLS : PRINT AT VAL "10",VAL
      "5";"Finished": STOP
9000 LOAD d**b$alpha" DATA b$()
9005 LOAD d**b$printC"CODE 65400
9020 LET L=1: LET o=0: GOTO L
```

The program is set up for an Epson printer with DISCiPLE interface, so users of other systems will need to customize. You will also need to alter the filename in the load instruction in line 9000 if you have used "b$mod1/2".

I am sorry that it has been necessary to use VAL "number" fairly extensively in this program but memory space is very tight even with the code in the UDG area. Those of you who have read the articles SPACE SAVING IDEAS in volume 2, issues 3 and 4, also by Clyde Bish, will know that whilst this causes little extra typing, a great deal of space is saved, particularly in a program like this where there are so many numbers.

By eliminating the screen prompts I suppose I could have put the code on the screen again but this seemed like working in the dark. I have already stripped out some refinements including the header. I suppose this could be included if screen prompts were omitted but it is easier to print a header independently.

I've already mentioned that L and o replace 1 and 0 in some places, this is a further space-saving feature but as it slows basic down considerably I have managed to avoid its use in the actual printing lines and in this way basic just keeps ahead of the printer, which is not very quick in Condensed mode.

As you will almost certainly need to do some customizing of this routine, some explanations may be helpful. If you wish to change the number of columns printed then you will need to change the code of "b$printC" also but this will be an interesting exercise for those like me, who are 'feeling their way' in assembler.

Line 12 finds the start of b$, adding only 6 to VARS as b$ is the first variable in memory. Lines 20 and 30 set up the 'routine variables' with the page-length going into 65390. If you need to alter this, change the '60' in Line 1510 to the same value.

I used a page length of 5 with b$ holding only 20 entries whilst I was developing these routines and still got through a couple of trees. Line 100 holds printer interface commands, setting right margin to 132 and telling the interface that a printer command follows. Line 110 sets Condensed mode - most printers will understand this, and Line 120 restores the interface to sending text. Modify as necessary.

Line 1001 tells you what is being printed and, if it is a full page, skips lines 1002 to 1005 which calculate the number of lines in the last (i.e. less than full length) page.

Lines 1010 to 1030 set variables from the values found by the code and the rest is a rather messy print routine dealing with each column in turn. Lines 1510/30 handle the arithmetic for subsequent pages calling the code from a different address to avoid the set-up instructions at the start.

Well, the end is here, this mini series comes to its close. I hope that these routines have been interesting or even useful to *FORMAT* readers (even if you are not using *Fastfile*). I also hope that the series will give inspiration to others to develop enhancements for this excellent little filing program.

## Writing for FORMAT

We try our best to get a good mix of articles together for each issue of *FORMAT* - but we always need more. Filling these pages is not easy, we start with a pool of articles and try to pull out a reasonable mix of Spectrum, SAM, and general purpose computing articles that we believe will appeal to the wide spread tastes of our readers. To judge from your comments we normally succeed.

However, we need your help, and we need it now. Sending us £50 notes would be nice, but seriously we do need your help. We need more material to fill these pages. Even some small items for Short-Spot and the Letters Page are better than nothing - and there can't be anyone out there who does not have something they want to say.

Larger contributions, from one page to a regular column, are also needed - urgently. Articles on computer languages, how to program, useful routines, listings, games hacking, hardware, experiences with printers.... There are just so many things that other readers will find interesting that you could write about. We can accept text in any wordprocessor format on DISCiPLE, PLUS D and SAM discs or from any PC wordprocessor on IBM discs (DD or HD).

*FORMAT* is your magazine, it keeps a very dedicated bunch of enthusiasts together and helps to keep their hobby alive. The Spectrum and SAM (and other Z80 machines) need *FORMAT*, and we need your contributions to help keep *FORMAT* out there in front - where it has been since 1987.

It will not cost much to send in a short piece for *FORMAT* and at the very least you will see your name in print and get a few months added to your subscription.

# VARIABLES
## *ON A THEME*

By:- Dilwyn Jones.

### Part 5.

Here we are again, happy as can be (I hope) and ready for yet another look at the Spectrum's system variables.

#### • 23679 POSN

Contains information about how far across the printer buffer the LPRINT position has got to. Contains (33-column number) for columns 0 to 31.

You cannot change the LPRINT position by POKEing this alone.

#### • 25680
#### • PRCC

Contains the low byte of the address where the next character is to go into the printer buffer, ie. this will contain (23296 + LPRINT column number), being zero for the left column of the printer, 15 for the 15th column, etc. Because this is the address of the top row of dots of each character, you can POKE this to change the LPRINT buffer position provided you change the value in P POSN (23679) to match. It may appear to work if you don't do this, but problems will be encountered at the end of the line.

#### • 23681
#### • Unused system variable

This system variable, although strictly speaking unused, usually contains 91. This is the high byte of the LPRINT buffer address (91*256 is 23296 where the buffer starts). This can be POKEd for your own use but using the printer will overwrite it and reset it back again to 91. 23680/1 together contain the address of the LPRINT position in the printer buffer. You will now affect the working of the printer if you POKE 23681, but anything stored here may be over-written by the printer routines.

#### • 23677/8
#### • COORDS

23677 is the system variable that contains the x co-ordinate of the last plotted point. After CLS it starts off at zero and 23678 is the system variable that contains the y co-ordinate of the last point plotted. It contains the actual value, so if the last point plotted was 3,3 both bytes would contain threes.

These two can be POKEd with valid x and y co-ordinates respectively. Since POKEing these does not actually PLOT anything on the screen, this is a convenient way to move the PLOT cursor around. This could be done by:-

PLOT OVER 1;X,Y:PLOT OVER 1;X,Y

but would be messy. Amongst other things this could simulate MOVE found in other Basics - useful if you wanted to draw lines from around a particular point.

#### • 23684/5
#### • DF CC

The address in the display file of the PRINT position. It may be POKEd to send the PRINT output elsewhere, although this requires an understanding of the way the display file is organized and I do not have space to cover that here. However, with a little work, you should be able to make use of this one.

- 23688/9
- SPOSN

23688 contains information concerning how far across the screen the PRINT position has got. It starts off as 33 for the left-hand side of the screen and decreases by one every time the PRINT position moves one place to the right. After using PRINT AT Y,X; (if Y and X ar valid PRINT AT co-ordinates) 23688 would contain 33-X. This can be useful when trying to prevent words being chopped in half when printed on the screen. If you imagine the number in 23688 as counting down towards zero as there is no more room on the current line, you can see that comparing this to the length of the word to be printed gives us an idea of whether it is necessary to move to a new line to prevent the word being chopped. Suppose the word to be printed was W$:-

        IF PEEK 23688<LEN W$+1 THEN
        PRINT

This only works for words less than 32 characters long.

23689 contains information relating to how far down the screen the print position has got to. It starts off at 24 for the top line of the screen and decreases by one every time the PRINT position moves down the screen. If you do not want a scrolling display and would rather the screen was cleared when the PRINT position got near the bottom of the screen, then try:-

        IF PEEK 23689=3 THEN CLS

- 23692
- SCR CT

Contains the data for how many scrolls will be carried out plus one before waiting with the message 'scroll?' to give viewing time. In graphics games

especially this can be a nuisance, since one is not interested in waiting for viewing. So, if the number in 23692 is anything other than one, the waiting does not occur. So POKE 23692,255 would give you 255 lines of printing before the machine waits with 'scroll?'

POKE 23692,0 seems to have a similar effect except that you have one more line of print. If more is needed, then if the printing is done within a loop, it will be necessary to include the POKE statement in the loop as well - time-wasting but necessary

- 23693
- ATTR 0

Contains permanent attributes, or the attributes (FLASH, BRIGHT PAPER and INK) in effect globally Local colours in PRINT statements, etc, are dealt with elsewhere. Note that most of the ROM routines use the values of the system variable holding the temporary attributes as these contain the permanent attributes unless a local parameter is specified. CLS, however, clears screen to the colours, etc, in ATTR P. The functions of the individual bits are -

Bit 7 is one for FLASH 1
Bit 7 is zero for FLASH 0
Bit 6 is one for BRIGHT 1.
Bit 6 is zero for BRIGHT 0.
Bits 5, 4 and 3 contain the PAPER colour in binary, eg. for PAPER 7, bits, 5, 4 and 3 would be 1, 1 and 1 respectively.
Bits 2, 1 and 0 contain the INK colour in binary, eg. for INK 3, bits 2, 1 and 0 would be 0, 1 and 1 respectively

Attributes of eight or nine are not dealt with here. If the permanent attributes are eight or nine, then those stored in 23693 may not be valid

- 23694
- MASK P

This is the system variable that helps the Spectrum determine the attributes of anything printed when a parameter of eight is specified. So, if you specified BRIGHT 8 globally, bit 6 of 23694 would be set to one to remind the computer in future that BRIGHT 8 has been specified. So, to determine the colour/flashing/brightness when printing, the computer looks at what's already there and prints the word in that colour, etc. Or if you like, it only overprints the character on the screen and leaves the attributes alone

Bit 7 is one when FLASH 8 is in effect.
Bit 6 is one when BRIGHT 8 is in effect.
Bits 5, 4 and 3 are normally all one when PAPER 8 in effect, but see below.
Bits 2, 1 and 0 are normally all 1 when INK 8 in effect, but see below

When there is more than one bit to consider, as in INK and PAPER, then only the bits set have their attributes bit taken from the screen. This can lead to some unexpected effects. Try this:-

        10 INK 8
        20 POKE 23694, BIN 00000011
        30 PRINT AT 0,0; INK 5;"555555
           5"
        40 PRINT AT 0,0;"1111"

As INK 8 is specified, you may only expect the ones to be printed in cyan like the fives, but no. Rather than check the INK attribute as a whole, it only checks the bits set in 23694, which were bits 0 and 1 See if you can work out what colour the ones will be printed in. Have fun!

- 23695
- ATTR T

This system variable contains the

current temporary colours as would be set up by local statements within PRINT statements. You could see this for yourself with something like these two direct commands -

PRINT PEEK 23695
PRINT INK 7;PAPER 0; PEEK 23695

That is, include the PEEK in PRINT statement under the effect of the local colour controls. Normally, unless local colour statements are specified, this system variable will contain the global colour values. Colours, etc, to be used for printing on screen are taken from these temporary system variables and things are balanced such that ATTR T is only different from ATTR P if local colour attributes and so on so decree.

- 23696
- MASK T

This is rather like MASK P (system variable 23694) except that the parameters here are temporary Normally the same as the equivalent permanent parameters, this is changed while local colours, etc are in effect. You could study this by using something like -

PRINT PEEK 23696, INK 8;PEEK 2369
    6,
INK 0; FLASH 8; PEEK 23696

- 23697
- P FLAG

This system variable contains, as you might expect from its name, flags used during printing. After PAPER 9 has been specified, bits 6 and 7 are set to one. After INK 9 has been specified, bits 4 and 5 are set to one. After INVERSE 1 has been specified, bits 2 and 3 are set to one. And after OVER 1 has been specified, bits 0 and 1 are set to one. The effects are global if the odd numbered

bits (bits 1,3,5 and 7) are set to one, and temporary if the even bits (bits 0,2,4 and 6) are set to 1.

### ◆ 23681 & 23728/9

These three bytes in the system variables are not normally used by the 48K Spectrum - you may like to make use of them as 'custom variable' for use in your own programs in which you need to access information. These are particularly useful in machine code routines where you can simply access the information via an address rather than searching for the variable in the variables area. 23728/9 was intended for use by non-maskable interrupts but these don't occur on the bare Spectrum.

### ◆ 23730/1
### ◆ RAMTOP

This two byte system variable points to the last byte of RAM of the Basic system area. Note that this is not the end of the memory used by Basic, in the sense that the user-defined graphics normally hide up above this address. If you move RAMTOP up above the start of the user-defined graphics they may be overwritten, but you gain quite a few valuable bytes which may be useful for 16K users.

One important thing is that NEW only operates as far as the address held in 23730/1, so you can store data above this which may be passed between programs loaded into the computer. The same is true if you want to preserve machine code routines, etc.

### ◆ 23732/2
### ◆ P RAMT

This contains the address of where RAM ends on the Spectrum. If you acquire a Spectrum whose memory capacity you don't know, then don't

bother looking inside it to see if it's an expanded model or not, just enter this expression:-

```
PRINT PEEK 23732+256*PEEK 23733-
16384
```

The 16384 bytes subtracted is for the ROM since RAM starts at address 16384 and goes up to the address held in PRAMT

And that brings us to the end of our look at the Spectrum's system variables. I have not covered the special ones set up for the 128K mode on later machines, nor have I looked at how interfaces (like the PLUS D/DISCiPLE) use system variables in special ways. When I first wrote the book on which this series of articles was based there were no such beasties around. Hopefully their omission from these articles will be rectified by someone in due course.

*... ut that's its #0. The price ticket is on the right hand side*

# The HELP PAGE

Edited By - Ray Bray.

It is good to be back with another Help Page this month although the number of direct questions we have received since the last Help Page has been minimal, therefore I am making up the space this month by picking up a query which was raised recently in another section of *FORMAT*

For the past five months I have been wondering how long it would be before we would receive the first question on SAM C, well it has now happened! Actually the problem concerns screen paging rather than SAM C itself J W Canfield, who lives in Bremen, Germany, has written a SAM C program to transfer data bytes for a sprite to the screen buffer, and then to display it using the PUT function. The routine to transfer the bytes to the buffer (shown below), plus the data bytes, are located in the top half of the memory. The screen is paged into the lower half (blocks A & B) whilst the transfer takes place, and is then paged out. Despite trying various modifications to the routine, it appeared not to be transferring the data to the screen pages although, if another page was substituted for the screen page, it operated perfectly. Mr Canfield asks what is he doing wrong? The listing of the routine, with my annotations, is as follows -

```
page1(int a,unsigned b,unsigned
c)
{
#asm
pop af
pop bc (number of bytes in C)
pop de (destination address in
DE)
pop hl (page number in L)
push hl
push de
push bc
push af
dec hl (decrement page number)
in a,(250)
ld (65535),a (save existing LMP
R value)
ld a,1 (page number)
set 5,a (set low memory to RAM)
res 6,a (set high memory to RAM
)
res 7,a (cancel write protect)
out (250),a (page-in screen pag
es)
ld hl,60015 (start of data)
ldir (transfer data)
ld a,(65535) (retrieve original
LMPR value)
res 6,a (ensure high memory set
to RAM)
out (250),a (page-out screen, p
age-in ROM)
ld hl,0
#endasm
}
```

The above routine was called with - a*page1(30,24576,192) and it was anticipated that the screen buffer would start at the absolute address 516096.

Looking at the routine there is nothing wrong with the paging sequence in general, it will page-in to blocks A & B, carry out a transfer of the bytes and page the ROM back in. However, there are two errors concerning the addresses used for the screen. Page number 30 is entered correctly as a parameter but then the value of HL is decremented which causes page number 29 to be paged into block A, with the screen (page 30) in block B. In modes 3 & 4 the screen is 24K long, therefore only the first part

of the screen is paged-in and the buffer is not present. The data bytes are thus transferred to the screen, not the buffer. The other addressing error is that 24576 has been used as the offset for the buffer when instead of 24577. If the former offset is used the first byte of data will be lost.

In summary, if you delete the DEC HL instruction and change the offset to 24577, then you will find that all the bytes will be transferred to the buffer starting at absolute address 532481, (not 516096). If it is only intended to write to the current screen then it is simpler to obtain the screen page number by including the instructions IN A,(VMPR) AND 31 instead of declaring the page as a parameter and using LD A,L. However your routine does enable you to write to any screen whether it is displayed or not. One general principle to be observed, (although it is not always possible to do so), is that it is simpler to have the operating routine in low memory and do the page switching in high memory, especially with the screen pages.

Following on neatly from this question is a point raised by P.J.Williamson in his informative article on SAM MIDI in the August issue of *FORMAT*. He stated that he didn't know how to use the SAM paging system and he thought that this had not been covered in detail in *FORMAT*. In fact the subject was covered to a large extent in Carol's excellent series 'Machine Code Without Tears'. However, I think your particular problem, where the Basic program is so large that it spreads into Blocks C & D of the memory, hasn't been covered, so perhaps I can throw a bit more light on that aspect.

Your requirement is the ideal case where you wish to switch pages in the high memory Blocks C & D. I don't intend to cover all the ground again concerning the paging registers and layout of memory, suffice it to say that two pages are switched at one time, and if you wish to switch (for instance) page 10 into the last block of memory (Block D), you have to ask for page 9 to be switched using the following commands:-

```
IN A,(251) ;Get the current val
ue of the HMPR
LD (STORE),A ;and store it some
where.
LD A,9
OUT (251),A ;Page-in 9 & 10 to
C & D respectively
CALL ROUTINE ;Your own M/C rout
ine.
LD A,(STORE)
OUT (251),A ;Restore original p
ages.
RET
```

Now with a large Basic program the problem is where to put this paging routine. One possibility is to put the routine in the System Heap, but you then have to ensure that you will have enough room for all the other uses of the Heap; the other solution, which I prefer, is to include the code in a REM statement in the first line of the Basic program. Putting code in a REM line can cause a problem on LISTing but if you always LIST from the second line you can get round it. In this instance the REM line lists without problem. The REM line in question has to be entered initially with a number of characters, the number being the same or greater than the length of the paging routine, in this case 20. In operation the paging routine is called from the second line of the program, which is a subroutine, to which the required page number is passed as a variable. The two lines will initially look something like this:-

```
10 REM XXXXXXXXXXXXXXXXXXXXX
20 POKE 23776,PAGE: CALL 23770:
   RETURN
```

The code for the paging routine is then poked to line 10 by entering the following lines and keying GOTO 30:-

```
30 CLS: FOR A=23770 TO 23789.
   READ C: POKE A,C: NEXT A:
```

```
LIST:STOP
40 DATA 219,251,50,236,92,62,
   0,211,251,205,X,X,58,236,9
   2,211,251,201,0,0  (NOTE.
   Put the address of your  o
   wn routine at X,X)
```

Now DELETE lines 30 & 40 before entering the rest of the program. If you are adding this to an existing program the program will either have to be renumbered to accommodate these lines or, if the first two lines can be squeezed in at the start, then the contents of lines 30 and 40 can be placed elsewhere.

If you wish to CALL a number of different routines when various pages are switched, then line 20 can include a DPOKE 23783,ADR statement. This will insert the address of the required routine into the paging code, the value of ADR being passed to line 20 along with the variable PAGE. I think you'll agree that this is a painless method of switching pages to access M/C routines from the Basic program. That is all I need to say now about this subject, apart from mentioning that all the pages are not available for your use as some are committed to the operating system, screens, ROM1, MasterBasic etc, depending on how your computer is configured. The best way to find out which pages are available is to examine the Page Allocation Table which is at 20736 to 20768. One byte is allocated to each possible 16K page and the simple rule is that if a page is not being used it is marked with a zero byte. For full details of page allocation see page 58 of the SAM Technical Manual.

This month we received a question which is somewhat different in that it is not exclusively to do with computers, however to demonstrate that your resident team are nothing if not versatile, we shall attempt to answer it to the best of our ability! S.M.S.Kempees writes from Rotterdam to say, "For some time now I have been wondering why on my keyboard there are two keys with a high point on them, keys F and J. Is this normal and what is the point of them?" (Was that an intentional pun!). No doubt many users have seen these high points and probably thought they were just a defect in casting the key, but I assure you they do have a useful purpose. They are in fact put there to assist touch-typists to locate their hands correctly on the keyboard and I would think that for visually impaired audio typists they are essential.

On the keyboard the centre row of the alphabetic keys is known as the 'HOME' row, meaning the row of keys over which the fingers are normally located. In this position the left and right hand index (1st) fingers are located over keys F and J respectively. Note the word OVER, as the fingers should not rest on the keys otherwise spurious letters might be entered. In this position the hands are positioned correctly for all the fingers to reach the rest of the keys. For the aspiring touch-typist I will take this just a little further.

In the HOME position the 2nd finger of the right hand should be over K and the 3rd over L, whilst the little finger is ready to operate the RETURN key. For the left hand the 2nd finger is over D, the third over S and the little finger over A. The SPACE bar is operated with either thumb depending which is most comfortable and the SHIFT key is operated by either little finger. From this basic position the fingers operate all the other alpha-numeric keys as shown in the diagram given in fig. 1 (next page).

Other symbols do not have a standard place on the keyboard, particularly on computers, so you will have to decide which fingers are best to operate these keys. The only other points I would like to mention in this mini touch-typing tutorial are that for comfort you should keep the wrists flat and use the fingers to depress the keys, not the wrists or forearms. By the way, the SAM keyboard
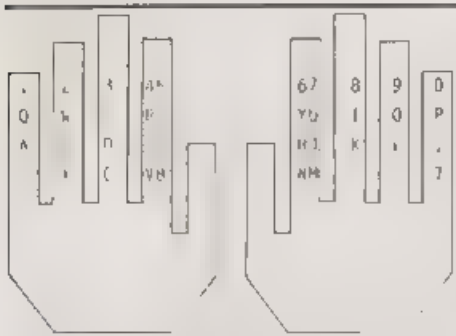
Fig.1 - Hands On Keys.

is very helpful in this respect because the wrists can be rested on the disc drive housing extension. Finally speed is not achieved by hitting the keys faster and harder, but by reducing the length of pauses between pressing consecutive keys. This can be achieved by practicing what is called CHAINING, however this is another aspect of touch-typing which you will have to follow up using a textbook.

With reference to whether or not these 'high points' on keys F and J are normal, all I can say is that I have regular access to seven different computers, including two laptops, and not one of them has this feature. I have also made a quick survey of electronic typewriters in the local stores, and out of ten different models examined, only one had such keys, so I would conclude from this it is not all that common. Incidentally Jenny has them on her keyboard and she says she couldn't do without them.

Finally, Ivan Betts has written from Belfast in response to our answer to D Birchall's problem with the SCART lead which appeared in the September issue. He writes: "..... Could the answer to his problem be as simple as - he has his lead the wrong way round? When you buy these leads there is nothing to indicate that it will only work one way round. I have my SCART plugs labelled SAM and TV and the odd time I don't

look and get it the wrong way round, I get a broken picture which is scrolling very fast. I have a Hitachi TV. Even if this is not his problem, there must be other frustrated people who spend hours trying to get their leads to work when a simple message from West Coast would make life simpler. As you probably know better than me - if there is a wrong way of doing something, you will find it before you find the right way. ...". Thank you for that tip Ivan, in fact it wasn't a possibility that occurred to me (RULE 1. Always look for the simple solution first, it usually saves a lot of time!)! My only excuse is that when I obtained my SCART lead, not from West Coast, it was marked clearly SAM and TV and I assumed that all manufacturers did the same. However if any supplier is now distributing unmarked leads I'm sure they will be pleased to take note of this plea and mark the plugs accordingly.

Well that's all for this month. May I make my own plea for you to keep sending us your queries, particularly on Spectrum and the +3. No matter how trivial you might think your problem might be, there is bound to be someone else out there who is thinking the same thing about the same problem. So don't be shy, share your problems and queries (and answers) with other readers of *FORMAT* that's what we are here for. Please keep sending your problems/answers to the following addresses -

Anything SAM or General Purpose,-
   Ray Bray (*FORMAT* Help Page.,
   Spring Cottage, Bourne Close,
   Porton, Salisbury, Wilts, SP4 0LL.

Anything +3, CP/M -
   Mike Atkins (*FORMAT* Help Page),
   70, Rudgwick Drive,
   Bury, Lancashire, BL8 1YE.

Please remember that if you want any discs/printouts etc returned then you must include an SAE

# EMULATION Exchange

The New Column For Emulator Users     By:- Peter Collier.

Over the last 5 years, there has been a slowly growing trend in computing. Old slow computers have been gradually superseded by the powerful desktop machines and super-consoles that we have today. As the technology has increased, people have welcomed it with open arms. What people have been slower to adapt to, is the need to keep changing software every time you change computers. This article is being written on a powerful 486 PC, but it could have just as easily have been written using Tasword on one of my Spectrums. As the technology has increased in power, and dropped in cost, so the possibility of using old software has seemingly become more of a problem. However, the increased computing power has also helped to solve the very problem it has created. The solution? Emulation!

For those that don't know, the principles behind emulation are very simple. Think of the differences between two computers, let's say a Macintosh and a Spectrum. The Spectrum has an 8-bit Z80 microprocessor. The Mac has a 16 or 32 bit 68000. The Spectrum has a simple 8 colour display. The Mac can have anything from monochrome up to True-colour (16 million colours). The Spectrum has, at best, simple 3 channel sound. The Mac is capable of high quality stereo digital sound. All of these factors are different (and more besides), but one common feature is that the Mac is more powerful in each respect than the

Spectrum. This means emulation may be possible.

It is generally acknowledged that for full speed emulation to be performed, the emulating machine needs to be about 10 times faster than the original. In practice, it is a lot more complicated than this. For example, the SAM can emulate a 48k Spectrum quite easily, but it isn't 10x faster. The reason here is that the SAM uses the same microprocessor (Z80) as the Spectrum, plus a similar graphics system, designed for that task. As the emulation software doesn't need to convert the programs to it's own processor language (as the Mac would have to translate Z80 machine code into 68000 machine code) or worry about re-directing the graphics (again the Mac would have to map each pixel onto an element of it's own display, and alter it's own colour map to accommodate the different colour modes). For a Mac to emulate a Spectrum, the job is much, much tougher.

Imagine you have a program written in Spectrum Basic. You want to run it on a PC, but the PC only has languages such as C or PASCAL available. You would have to take each line of program and re-write it in the PC language, converting both the keywords used, and the syntax associated with them. This is the problem in emulating one processor on another. A 68000 will need to convert each machine code instruction into one of it's own, also converting memory

addressing and flag manipulation. All of this in real-time. It takes a lot of speed. Add to this that the graphics, sound, I/O, disc access and keyboard scanning all need to be done AS WELL.

The current generation of home computers are now capable of emulating the older 8-bit machines. There are emulators available for the Spectrum, ZX81, C64, Atari 800, Amstrad CPC, Dragon 32, Tandy colour computer and many more too. The emulating machines vary, but are generally most successful on the real powerhouse PCs and Apple Macs. The advantages of emulation are obvious. Transfer your old data from one machine to another without having to alter it. Run software that you are familiar with alongside the latest power systems. Archive your old software and files on newer, more reliable systems. Let's face it! It must be better to keep your Spectrum programs on a virtual microdrive file on a PC hard disc than to keep on using the tired old tape time after time after time, praying that it doesn't snap or get caught up! I find this is one of the God-sends of emulation. I have a room full of old computer cassettes and discs of weird origins. But, now my entire Spectrum collection is backed up by copies on neatly archived PC discs. I have about 2000 Spectrum programs, archived onto about 15 PC discs. I still use the originals, but if

something goes wrong and the tape gets chewed or just wears out, I can still use all of my software. I can even download it from the PC back onto a new cassette! I have just done this recently. It took me an hour, but that's just fiddling about with wires and trying to get the tape volume right.

Spectrum emulators were some of the first to arrive on the scene. They are now one of the widest ranges of emulators available of any computer. Versions are available for the PC, Mac, Amiga, ST, Acorn Archimedes, UNIX and QL computers. Some are better than others. Some are more realistic than others. Don't bother trying to run a Spectrum emulator on an Amiga 500 or 286 PC. Although the emulators will run on these machines, they are too slow to get very close to 'real' Spectrum speed. On a 486 PC, however, you will need to slow the emulator down because it will work at many times the speed of a 'real' Spectrum. All of them are similar in operation. They all have copies of at least one Spectrum ROM image. (This is a binary copy of what is on the Spectrum ROM chip.) The ROM was partially released by Amstrad a few years ago, with the agreement that it may be copied and used in emulators, but it may not be sold and it must be clear that Amstrad maintain the copyright to the code. They all also have a Z80 processor built in software. These pieces of code I have great admiration for. I tried to emulate the 6502 processor used in the Atari 800 and C64 computers. It is simple compared to the Z80. Boy, is it tough to copy that logic into software! Some of these processor emulation's are so like the real chip that it is really quite amazing. They copy every register, every interrupt signal, every opcode (the

machine code 'keywords', if you like) and even the chips internal clocking. Compared with re-directing an RS232 stream to a physical port, the processor emulation is really tough.



Emulators also have additions that the originals never had. There were many drawbacks to the original Sinclair Spectrum: it was unreliable, got hot, took a long time to load software, had a rubber keyboard, not much in the way of communication sockets, etc. The emulators have the advantage that, whilst maintaining the character of the original machine, they can also enhance and improve on it's bad points. For example, the loading times. Using a real cassette, loading times were usually about 4-5 minutes for a 48k program. This was good compared with some of it's peers (I have a program for the Atari 800 which takes 45 minutes to load!) but it's still time to kill, and really frustrating if it fails at the last moment. Even microdrives are slow, and even more unreliable, particularly when they get old. The emulators all try to make some improvements to this. The most common solution is the snapshot. If you have used either a PLUS D/DISCiPLE or a Multiface, you may well be familiar with the idea. A snapshot is like a 'photograph' of what is going on in the computer at a given moment. Normally, they consist of the whole of the

computers RAM, the state of each of the registers and the program counter (where the program has got to in it's execution) being stored in a single file. The different formats all vary, but need to include at least these things. When the snapshot is re-loaded, everything is restored to the exact place it was before, and execution carries on as if nothing has happened. I actually used to use this idea to help with tough sections of games on my real +3. If I knew a tough bit was coming up, I would snap the game and save it to disc. If I got the game wrong, I could simply reload it to the point I had been a few moments before!



Some of the emulators have got quite advanced in their enhancements. Z80 has a menuing system to allow the options to be changed and things to be loaded and saved. It also allows virtual tapes to be stored on disc (as do some other emulators). These are either audio samples of the tape, or copies of the tape, stored in the same format as the Spectrum saved them. These '.tap' files can include all the features, such as loading screens, and can be rewound and forwarded, but they are limited to certain loading types. Speed loaders can't be easily adapted for them. These virtual tapes have the advantage that they can be loaded as quickly as the emulator can handle them. The drawback is that they do tend to take up more space than

snapshots. Another common addition to the emulators, is some form of I/O simulation. Most seem to use some kind of Interface I style RS232. This is re-directed to the host machine's serial port. Some also have the ability to load and save to/from tape. This is normally through either a sound card or the parallel printer port. Mice and joysticks are also popular. All of these additions both enhance the qualities of the machine Sinclair built, and make using an emulator as much a pleasure as the real McCoy

The advantages of using a good emulator can almost outweigh the ability to use a real machine. I say almost, because somehow there is a certain satisfaction from watching the blue and yellow bars during loading and the bizarre rituals which go with loading - creeping around the room, not touching the joystick and keeping the cassette player as close to the Spectrum as possible. The one thing the emulator can't copy is the 'feel' of the real machine. It's all clinically efficient and dependable. It's like Stirling Moss' F1 cars compared with Damon Hill's. The old one doesn't work as well, or as quickly or as reliably, but it does have a feel all it's own

Emulators really can be practical, and not just a novelty to show to friends. For example, if you load Z80 with a program like Tasword. The PC screen display is much clearer and easier to read than the Spectrum's. The files can be saved and loaded in exactly the same way as the real machine, but faster. The software itself can be loaded faster too. The keyboard is easy and fast to type with. You can print out the document to a laser printer as easily as you used to do it to the ZX printer. I have also tried to

develop programs using the emulator. This is a vast improvement. Put Z80 into 128k or use an additional language (such as Beta-Basic or HiSoft C or PASCAL) and typing programs in becomes faster and easier. Saving the program at regular intervals is also more convenient because you can just knock off a quick snap to the host machine's hard drive. Programs like compilers or assemblers work just as well, and can be speeded up on faster machines by pushing the emulator speed up. I also tried to do some graphics in Basic. You know, the kind of line drawings that you get in 'Learn to program' books because they look good, but in practice, they are horribly slow. On an emulator even Basic graphics can be fast!

In the next few articles, I will be taking you on a ride into the world of the Spectrum, circa 1990s style. The machine may have been out of production for some years, but the last couple have seen a strong revival. I will be demonstrating how to use the 'non-Speccy' parts of an emulator and proving that Sir Clive's little machine lives on in 1995. For now, though, here's a thought. I can run an emulator called XZX on my SPARC workstation at work, playing a classic Spectrum game like JetSet Willy, and if the boss comes along. I can press a button and the game hides as an icon at the bottom of the screen, waiting for me to bring it back into view. Now, that's a trick you can't do with a real Spectrum!

If you have any questions, or information, you would like to see included in this column, then please feel free to send them to me, Pete Collier, 90, West Street, Burgess Hill, West Sussex RH15 8PQ or e-mail them to me on

etjprtr@etjxdmx.ericsson.se

30
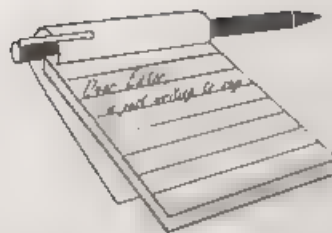
**Dear Editor,**

Ettrick Thomson's reference to his early work with the Ferranti Pegasus Mainframe computer makes a mention of the Computer Conservation Society (CCS) timely. Amongst several examples of the earliest production computers collected by the CCS there is a working Pegasus, several Elliot 803s, and many others.

The CCS is currently based at the Science Museum in London, with which it has a close working relationship, although a move to Bletchley Park, wartime home of the Enigma decrypters amongst others, is currently under consideration. Bletchley Park is now open to the public, and already houses a growing collection of early computers, many in working order, plus an excellent exhibition illustrating the code breaking activities.

The CCS is open to anyone interested in computer conservation and the history of computing, and membership is free. Amongst its aims is the conservation of historic computers and identify existing computers which may need to be archived in the future. From recent correspondence in the 'nostalgia' sub-directory, there may be a significant number of *FORMAT* members who had a hand in the operation, and who knows, the design or development of the earlier machines. The CCS Secretary is Tony Sale, 15, Northampton Road, Bromham, Beds. MK43 8QB.

In the not too distant future I would suggest that steps be taken to see if an *Uncle Clive's Corner* could be set up, with examples of the ZX80, 81s, Spectrum, SAM, etc., plus the far from inconsiderable range of add-ons, official and otherwise, plus at least the 'serious' software. *FORMAT* would seem the obvious, if not the only remaining organization to start the ball rolling.

Taking another track, Peter Wood of the Croydon Model Railway Society will be interested to know that he is not ploughing a lonely furrow - or should it be cutting - in producing his Society's Journal by SAM. My own Spectrum plus Discovery Disk Drive has been used not only for the admin of the Model Electronic Railway Group, but also the quarterly Newsletter and many of the more recent Technical Bulletins, which required the combined services of TASWORD 3, PCG's DTP package, and Artist II. Printed circuit boards have also been produced using PCB Designer, by Kemsoft of Worcester the quality of the resulting artwork impressed the firm I asked to etch the boards, although they shouldn't have been surprised as they had used a Spectrum to run their first production line!

The same equipment has been used until very recently to produce the Technical and Operating Manuals, as well as a number of PCBs for the two model railway exhibits at the Pendon Museum of Miniature Landscape and Transport at Long Wittenham, near Abingdon, Oxon (Tel. 01865 407365 for opening times.)

Last but not least, I have demos up and running on Spectrums (Spectra?) of Model Railway Operating Timetables, with multiple displays via Interface

31

One's and the Spectrum 'net', a 'Next Train' display for the viewing public, using a combination of DLAN, screen compression, and a customised slide-show program, and an on-screen simulation of a model railway to be built by a Finnish friend when he retires (optimist!). His face when he saw it 'up and running' was a picture, and he is now wondering if he needs to bother with the real thing!

On a personal note, the last mentioned involves the use of a 'Blockmover', routine first published in Outlet and also ARTIST II. I understand that *FORMAT* has the copyright for the PLUS D has the copyright for the tape or version. How do we stand for the tape or Discovery version, please? The 'Next' Train' display combines several routines, one of which is a nice 'iris' where the border collapses to the centre of the screen re-opening to reveal the next screen which has been extracted from memory, relocated, and decompressed to the screen in the interim. It also uses DLAN, Campbell Systems Display LANguage, which John Campbell generously released into the public domain when I rang him to ask about a small modification to pause the display until a designated key is pressed.

Sorry not on disc - the PC was fired up, and I'm writing this at 0630 hours before setting off for Quedgeley from deepest Sussex. Spectrums (seven of them!) still packed up after showing the foregoing at the big Model Railway Exhibition at the NEC last weekend....

*Yours sincerely, Paul King.*

Well Paul, lots to get the grey cells going in your letter. It was nice to talk to you at the show the other day.

Anyway. The main point you want an answer on is the Artist II question. Yes, we can supply the Artist II on 3¼" or 5¼" disc for DISCiPLE or PLUS D. We did offer the Discovery User Group a deal a couple of years ago to allow them to produce a version on Discovery disc but I did not hear back from them. As my Discovery is terminally sick I can't help on that one at the moment. The tape version is still copyright but, if you were only giving away the bare program (and not all the extra programs and files that make up the full package) I'm not going to come down on you from a great height. So, keep up the good work with the model railway stuff and see you at the next Gloucester show. *Ed.*

---

**Dear Editor,**

SAM C software has now been available from Fred Publishing since April of this year. Although SAM C has been reviewed by your experts, I have noticed a dearth of reaction from the readers. As I so far have not had much success in getting my software to function properly and nothing has come back to me from Fred, following my letter of complaint, except an acknowledgement of my letter, I am naturally wondering how other readers who have purchased the software have fared.

With a new language with new possibilities to investigate, I should have thought your postbag would be full of correspondence on the subject. Is there a conspiracy of silence on the subject or is it that the readers are just not reacting?

*Yours sincerely P.J.Williamson.*

We do have a couple of C articles coming along, the main delay being that I have to learn a little about C before I can edit C articles properly. *Ed.*

---

**Dear Editor,**

Just a letter from the sun to sunshine *FORMAT* !!!

You will notice that I'm writing on fax paper, and the reason is simple, I am using a serial printer 8056 purchased at Boots Stores some time ago for my Spectrum 128+ and now I have connected to an 8088 Olivetti M240 and

you can see, it works!!!

The word processor is a PC text - not very good, but good for my experiment. How can you connect this printer and Why? Well, the fax paper here costs only £1.00 per roll, about 1000 feet.

Hardware required:

The printer has a cable with 2 Black wires, 1 Red wire and 1 White wire.

Solder the 2 Blacks together to pin 7 of female 25 pin connector. Solder the 1 Red wire to pin 2 of the female 25 pin connector. Solder the 1 white wire to pin 5 of the female 25 pin connector. Join with a loop wire pin 4 and pin 20 of the female 25 pin connector.

And, you are ready to connect the printer to serial 25 pin male connector of any PC compatible.

Software required:

Add to your autoexec.bat file the following lines:-

1... Mode LPT1:=COM1:

2... Mode COM1:1200,N,8,1,-

This Redirect the command print. And now restart your PC for the new autoexec to execute these two lines.

Happy printing on fax paper!!!

P.S. I still used the Speccy with *FORMAT*.

*Yours sincerely, L.Senna-Cheribbo, (Spain).*

---

**Dear Editor,**

I would like to make a request for a Mr Andrew Watkins, (who's letter was published in Vol.8 N°12) to confirm as to whether his new SCART lead which he hoped to use for a Sony KV-M1400 TV was in fact a success, as I would like to arrange the same for my Spectrum 128 which also uses a Sony KV-M1400U TRINITRON and I'm certain quite a number of readers would really like to know too.

In page 3 of my Spectrum 128 introduction, I found a diagram which explains clearly enough the RGB terminal points. However, the real

nature of the problem seems to be trying to determine the exact order these correspond at the SCART end.

And so at this point, I wondered if either Andrew or Ray might send a diagram of the pin layout of the Sony KV-1400U TV (since this finer point of detail isn't shown in the above TV's instruction book).

The prospect of getting a greatly improved picture on the Sony are most encouraging, particularly as Ray points out that "a SCART equipped TV will give as good a performance as a monitor" and that he has been using this for the last two years with excellent results.

Many thanks to Andrew Watkins in advance, and indeed to Ray Bray who also helped to throw some light on this particular subject.

And last but not least of all, thanks to the team of *FORMAT* Publications who's hard work has helped to make all this possible.

*Yours sincerely, C.A.Walford.*

---

**Dear Editor,**

Just a short note, mainly to place an order, but also to congratulate you on the quality that *FORMAT* has reached over the last nine years.

As a subscriber from day one, I've seen many changes in the computer world; from the beginning as MGT to SAMCo and now West Coast Computers. *FORMAT* has always kept us informed and been packed with useful information, hints, tips and reviews. Thankyou!

Keep up the good work, and here's to the next nine years - who knows what they will bring?!

All the best for the future.

*Yours sincerely, S.Atkins.*

---

**Dear Editor,**

I have some questions to ask you about some products there are advertised in

the *FORMAT* magazine.

1. I want to upgrade my SAM Coupé with a second disc drive. There are some kits from West Coast Computers, but I understand that I have to remove connectors from the main SAM board. My question is: there are two connectors on the board, to fit two disc drives (I already have one), can I still use my existing drive if I buy one of the new ones or do I have to remove both of the connectors?

2. About the external disc drive interface, can it support any drive mechanism (eg, drives used by IBM)?

3. Some software needs MasterDOS and MasterBasic. Are they the last developments? How can I get them?

I only have SAMDOS 2 and ROM 2. How can I get an upgrade ROM?

4. I need to buy an emulator of the Spectrum 128K +3, does the program Specmaker run like one, or does is just run 48K software?

5. My SAM Coupé runs at 6MHZ, can I upgrade to 8MHZ or higher. How much higher? And how?

6. Some advertisers of software demand an SAE, to send a list of software. I don't know what that means. If there is enough money can you send to Revelation and to Atomik Software an SAE for me.

#### *Yours sincerely, Leonel Filipe Santos Barros, (Portugal).*

OK Leonel, here are your answers.

1) You can still use your existing drive, either as drive 1 or drive 2 when you fit a modern disc upgrade. These will work with most PC type drives so it may be cheaper for you to buy the drive locally and just order the kit of electronics from us.

2) The external drive interface (XDI) will work with and BBC compatible drive. This is the same standard as used with the DISCiPLE and PLUS D on the Spectrum. A drive, complete with power

supply is needed.

3) MasterDOS and MasterBasic are still available from BetaSoft, see their adverts in past issues of *FORMAT*.

4) The only emulators for SAM are 48K, there has been some effort to transfer some 128K games to SAM but they are done one by one, there is no way to emulate the 128K memory page switching without extra hardware.

5) No, the SAM will only run at its own speed. There are some attempts going on to build a board which sits on the back of SAM and allows faster running but that requires almost all of the internal hardware to be replaced by chips on the external board. This makes it expensive (£150 or more I have heard) so I don't think it will turn out to be a commercial product.

6) An SAE stands for Stamped Addressed Envelope. If you obtain IRCs (International Reply Coupons) from your local post office and send them instead then all the company has to do is hand over the coupon when it posts your letter to you.

I hope that covers everything, if not then drop me another letter sometime. *Ed.*

---

#### Dear Editor,

First and foremost, will you please see that Jenny gets a big pay rise this year for the way in which she dealt with my recent complaint.

Please find enclosed a cheque number to renew my membership with *FORMAT* for another year. I always despise doing this as I can never complete the renewal notice because there is very little I don't like about *FORMAT*, and enjoy every issue.

Keep up the good work and more power to your elbow!

#### *Yours sincerely, Bob Chowdhury.*

Thanks Bob, but its all part of the service we provide, and its always a pleasure to hear from another satisfied

customer. But, if there is anyone else out there who expresses a wish to see me financially better off then perhaps they can drop me a line. *Jenny.*

And if that line is written on the back of a £20 note then I just may be able to pay Jenny a bit more. Although I must point out that the pleasure of working with me should be reward enough for anyone. *Ed.*

### Dear Editor,

I am currently doing my GCSE's and for my technology project I want to build a device that will allow a network lead to be 12 metres with the signal still strong enough to work. I believe that if a network lead is over 2 metres long then it doesn't work very well and if it is over 3 metres then it doesn't work at all. It would be very helpful if you could send me as much information on the network/MIDI port as possible. I need information such as the speed at which it runs at, what type of signal is sent out, how strong is the signal etc. Also is there anyone else I could write to for information because for the investigation & research part of my project I need infomation from different people.

*Yours sincerely, Matthew Beaman.*

There was some testing done in the early days of SAM, back in the heady days before MGT launched the machine. On one occasion I remember sitting in the upstairs office at Lakeside transfering files up from Bruce Gordon's SAM which was downstairs. That lead must have been at least 30 feet long and everything worked. However, I've never tried it with a production machine. Anyone out there able to help Matthew with his GCSE project? We will pass on all letters we receive. *Ed.*

### Dear Editor,

I would firstly like to thank you and Jenny for your help over the last week or so. I know I must have been a bit of a

pain over the last few weeks. I look forward to being able to thank you both in person at the Gloucester Show.

I would like to draw your attention to a mistake in the SAM manual and the Technical Manual, where two diagrams completely contradict each other. One diagram says that pin 20 of the scart connector is bright TTL and pin 12 is +5 volts, and the other says that pin 12 is bright TTL and pin 20 is +5 volts. For the information of other readers, pin 20 is the bright TTL connection.

I can send an article about Spectrum emulators if you like, and while we are on the subject of the Spectrum, are there any mail order companies who still sell Spectrum titles such as Dizzy?

Thank you both once again, I shall go now and get off your back for a while until my curse strikes again and something else goes wrong with either SAM or my +3.

*Yours sincerely, Stewart Skardon.*

Thanks for the information on the Scart connections Stewart, and it was good to put a face to the voice at the show last week. And yes, we would welcome anything you would like to contribute to these pages. *Ed.*

### Dear Editor,

I have noted with interest the SAM C articles written by Carol Brooksbank, long may they continue.

I would like to put a plea to Carol Brooksbank. I was hoping to write a faster program of "Spirograph" using SAM C but SAM C does not yet have a Maths package.

I require to use SIN and COS. I noticed that in Carol's Circle program that she uses a ROM call. Is it possible that she could in a future C article deal with my problem?

*Yours sincerely, Frank Evans.*

We can but ask her Frank, over to you Carol. *Ed.*

### Dear Editor,

I feel I must write to agree with Ettrick Thomson's comments regarding SAM C in last months *FORMAT*.

Fred Publishing are incorrect to advertise SAM C as a C compiler: it is in fact a compiler of Small C - a subset of 'real' C which omits, among other things, floats.

Having said this, the only other Spectrum/SAM implementation of C (Hisoft C) also omits floats, and it was for this reason that I stayed with Hisoft Pascal and its superb maths routines for writing Fractal Collection. I would recommend using Hisoft Pascal for any maths intensive software.

*Yours sincerely, Daniel Sheridon.*

SAM C may not be ANSII standard C but it is certainly C. Floats will come in time, and as people get round to using C more they will find that there is nearly always more than one way to crack a egg. And at the price you are getting very good value for money. *Ed.*

### Dear Editor,

If I wanted to buy a SAM Coupé from you (of course, using your Foreign Currency Payment Scheme) How much extra would it add to the price of the SAM (in this case, the W1010), or, in other words, how many pounds more or less, would it take for you to post it (even if you didn't use airmail service, if you use normal posting).

Thanks for the attention.

*Yours sincerely, Candido Antonio Esteves Madaleno, (Portugal).*

The extra postage for a single drive SAM Elite costs £15 while a dual drive would cost you £20. For Europe there is now only one rate. Hope this is the info you need. *Ed.*

### Dear Editor,

Frank Evans (letters, July 95) brings up the question of a modulus function in

SAM C. As you say, there is one: n%m is supposed to give n modulo m. BUT this function does not always give the same result as n MOD m as used by mathematicians, in which m is positive, n can be of either sign, and n MOD m is in the range 0 to m-1; in Pascal, and in SAM Basic, MOD conforms to this convention, but in SAM C, n%m has a negative value if n and m are of opposite signs.

Here is a SAM C function, mod (n,m), that gives n MOD m as in SAM Basic:-

```
int mod (int n, int m)
(int r; r=n/m;
if (n%m<0)r--;
return n-m*r;
)
```

This is the equivalent of SAM Basic n-m*INT(n/m).

*Yours sincerely, Ettrick Thomson.*

Many thanks Ettrick. *Ed.*

### Dear Editor,

I see from the latest issue of *FORMAT* that West Coast are dropping the TV modulator versions of the SAM Elite. Surely this is a bit daft as it cuts their potential market. I would have thought that the need was to get as many SAMs sold as possible and not everyone can afford a monitor to use with their home computer.

What do you, and other readers feel about this move?

*Yours sincerely, Barry Marks.*

I understand that the move has been prompted by the shift of sales away from the TV version since they started selling the Scart version earlier this year. The screen picture is just so much better with a monitor and most people seem to recognize this.

However, all is not lost. There are plans for a new Modulator Unit, due out in the new year, that will plug into the Scart socket on SAM and give a much

better TV picture than the existing modulator that is buried in the power supply. *Ed.*

### Dear Editor,

Is there anyway that I can obtain a new Slim-Line drive for my SAM Coupé? Being overseas it is too expensive to send my SAM to you for the fitting of a new type drive and I've never used a soldering iron in my life. What can I do?

*Yours sincerely, R.D.Boon, (Germany).*

Well R.D. (I do wish people would give their first name in letters like this) there are no more NEW slim-line drives. They are not available in the UK anymore. However, West Coast Computers do sometimes get a few reconditioned ones which we normally sell at shows with a 90 day warranty. If you want one we can put your name down on a waiting list. The cost to Europe would be £65 including postage. *Ed.*

### Dear Editor,

Should you still require a programmer to convert the BBC software to the SAM. I will be available. I can also build and test prototype hardware, having the experience of working in the electronics software, but do not have the time to design new hardware.

*Yours sincerely, Tim Wells.*

I would love to talk to you on both the software and hardware front Tim, but you did not give a phone number on your letter. Please give me a ring some time soon and we will have a chat. *Ed.*

Letters may be shortened or edited to fit on these pages although we try to edit as little as possible.

This is YOUR letters page so it is up to you to fill it with interesting things. Come on, get writing, any subject even remotely related to computers. Just keep things as short as you can so we can fit in as many as possible each month. Please write clearly or type your letters. Send them to the address on page 3 or fax them to us on 0452 380890.

---

COURT FLASHER

---

## SOFTWARE

### SAM Coupé Software On 3½" Disc

**PCG's DTP PACK.** The famous package including WordMaster, TypeLiner, HeadLiner. Complete on disc for SAM. R.R.P. £37.50. *SPECIAL READER'S BONUS* - Now includes 2 extra font packs worth £15.90 bringing the total pack value to £53.40.

Order Code: SCD-04      Price: £24.95

**COLOUR DUMP.** Screen dump software to work with Epson compatible colour printers. *>>New Upgraded Version<<* RRP £12.95

Order Code: SCD-05      Price: £7.95

### SAMSPEC SOFTWARE

SAM + Spectrum versions on one disc (5¼or3½

**FONT LIBRARY.** 100 screen fonts plus support software, easy to use on either computer.Special +3 disc available (Spectrum files only). RRP £8.95

Order Code: SSD-01      Price: £5.95

**MONEY MANAGER.** Advanced personal budgeting system for both machines. Spectrum version needs 128K/+2 (not +2a or +3). RRP £15.95

Order Code: SSD-02      Price: £9.95

### SPECTRUM SOFTWARE

Unless otherwise indicated these are DISCIPLE/ PLUS D programs. Please state 3½" or 5¼" disc (80 track only) on order.

**ARTIST 2.** The best art package ever written for the Spectrum. 48K and 128K versions on the same disc. With page-maker and utility software. Full manual. RRP £19.95

Order Code: SPD-01      Price: £12.95

**UTILITY DISC #1.** Six of the best and most useful disc utilities for DISCIPLE or PLUS D. Added bonus program - not to be missed. RRP £9.95

Order Code: SPD-02      Price: £6.95

**PCG's DTP PACK.** The famous package including WordMaster, TypeLiner, HeadLiner. R.R.P. £37.50. *SPECIAL READER'S BONUS* - Now includes 2 extra font packs worth £15.90 bringing the total pack value to £53.40.

Order Code: SPD-04      Price: £24.95
Special 3" disc version for +3
Order Code: SPD-04C      Price: £29.95

## FORMAT BACK-ISSUES

Volumes 1, 2 and 3 are no longer available. Volumes 4,5 and 6 are now available only as complete volumes.

Order Code: FBI-V4 (Vol.4)   Price: £12.00
Order Code: FBI-V5 (Vol.5)   Price: £12.00
Order Code: FBI-V6 (Vol.6)   Price: £12.00

All issues, from Volume 7 Nº 1 to the latest issue, are available seperately. To order just quote the Volume/Issue Nº. Price: £1.50 each, (5 or more £1.25 per copy).

---

## FORMAT READERS SERVICES

### FORMAT BINDERS

We are sorry to say that the range of binders we have been selling for a number years has now been discontinued.

We are urgently attempting to source a new binder and will bring you news of it as soon as we can. In the meantime, if any reader has a source for a suitable A5 binder we would love to hear from you.

### HARDWARE

(For SPI please see West Coast advert.)

**DUC LEAD** For use with DISCIPLE or PLUS D. This 'Dualing Up Cable' allows two disc drives to be attached without the need for opening one drive to alter its ID. Allows you to switch drives to even out wear without opening cases. RRP £16.95

Order Code: DUC-01      Price: £11.95

**PRINTER LEAD.** Top quality parallel printer lead (BBC standard). 2 metres long. For use with DISCIPLE, PLUS D or SAM. RRP £10.95

Order Code: PPL-01      Price: £7.95
+3 PRINTER LEAD. For +3 or +2a. RRP £12.95
Order Code: PPL-02      Price: £9.95

**SAM COUPÉ TECHNICAL MANUAL Version 3.** The real inside info on SAM. No SAM is complete without one. RRP £16.95

Order Code: STM-01      Price: £12.95

#### Music Maestro

Supplied on 3½" disc for PLUS D/DISCIPLE (3" for +3) this excellent program allows you to write music either just for the fun of it or to include in your own games software where it plays under interupts. Full manual, lots of demo tunes. The ideal way to write music on the Spectrum. R.R.P. £9.95

Order Code: SPD-05      Price: £7.95
+3 version: SPD-05C      Price: £7.95

ORDERING All prices include UK p&p, overseas readers please add 10% to for extra postage.

Clearly state Order Code, Product description, Quantity required and Price. Remember to add any postage and don't forget your Membership Number or we can't process your order. Payment in STERLING by Cheque (drawn on a UK bank), P.O. Euro Cheque or Cash. Make cheques payable to FORMAT. Payment MUST be in same envelope as order. Send to FORMAT's address on page 3. Normally dispatch is at the same time as your next issue of FORMAT. We will not be held liable for delays or non-delivery due to circumstances beyond our control.